# NAG Toolbox

# nag_contab_tabulate_stat (g11ba)

## 1 Purpose

nag_contab_tabulate_stat (g11ba) computes a table from a set of classification factors using a selected statistic.

## 2 Syntax

```
[table, ncells, ndim, idim, icount, auxt, ifail] = nag_contab_tabulate_stat
(stat, update, weight, isf, lfac, ifac, y, wt, table, ncells, icount, auxt,
'n', n, 'nfac', nfac, 'maxt', maxt)
[table, ncells, ndim, idim, icount, auxt, ifail] = g11ba(stat, update, weight,
isf, lfac, ifac, y, wt, table, ncells, icount, auxt, 'n', n, 'nfac', nfac,
'maxt', maxt)
```

## 3 Description

A dataset may include both classification variables and general variables. The classification variables, known as factors, take a small number of values known as levels. For example, the factor sex would have the levels male and female. These can be coded as 1 and 2 respectively. Given several factors, a multi-way table can be constructed such that each cell of the table represents one level from each factor. For example, the two factors sex and habitat, habitat having three levels (inner-city, suburban and rural) define the $2 \times 3$ contingency table

| **Sex** | **Habitat** | | |
|---|---|---|---|
| | Inner-city | Suburban | Rural |
| Male | | | |
| Female | | | |

For each cell statistics can be computed. If a third variable in the dataset was age, then for each cell the average age could be computed:

| **Sex** | **Habitat** | | |
|---|---|---|---|
| | Inner-city | Suburban | Rural |
| Male | 25.5 | 30.3 | 35.6 |
| Female | 23.2 | 29.1 | 30.4 |

That is the average age for all observations for males living in rural areas is 35.6. Other statistics can also be computed: the number of observations, the total, the variance, the largest value and the smallest value.

nag_contab_tabulate_stat (g11ba) computes a table for one of the selected statistics. The factors have to be coded with levels $1, 2, \ldots$. Weights can be used to eliminate values from the calculations, e.g., if they represent 'missing values'. There is also the facility to update an existing table with the addition of new observations.

# 4 References

John J A and Quenouille M H (1977) *Experiments: Design and Analysis* Griffin

Kendall M G and Stuart A (1969) *The Advanced Theory of Statistics (Volume 1)* (3rd Edition) Griffin

West D H D (1979) Updating mean and variance estimates: An improved method *Comm. ACM* **22** 532–555

# 5 Parameters

## 5.1 Compulsory Input Parameters

1: **stat** – CHARACTER(1)

Indicates which statistic is to be computed for the table cells.

**stat** = 'N'
The number of observations for each cell.

**stat** = 'T'
The total for the variable in **y** for each cell.

**stat** = 'A'
The average (mean) for the variable in **y** for each cell.

**stat** = 'V'
The variance for the variable in **y** for each cell.

**stat** = 'L'
The largest value for the variable in **y** for each cell.

**stat** = 'S'
The smallest value for the variable in **y** for each cell.

*Constraint*: **stat** = 'N', 'T', 'A', 'V', 'L' or 'S'.

2: **update** – CHARACTER(1)

Indicates if an existing table is to be updated by further observation.

**update** = 'I'
The table cells will be initialized to zero before tabulations take place.

**update** = 'U'
The table input in **table** will be updated. The arguments **ncells**, **table**, **icount** and **auxt** must remain unchanged from the previous call to nag_contab_tabulate_stat (g11ba).

*Constraint*: **update** = 'I' or 'U'.

3: **weight** – CHARACTER(1)

Indicates if weights are to be used.

**weight** = 'U'
Weights are not used and unit weights are assumed.

**weight** = 'W' or 'V'
Weights are used and must be supplied in **wt**. The only difference between **weight** = 'W' and **weight** = 'V' is if the variance is computed.

**weight** = 'W'
The divisor for the variance is the sum of the weights minus one and if **weight** = 'V', the divisor is the number of observations with nonzero weights minus one. The former is useful if the weights represent the frequency of the observed values.

If **stat** = 'T' or 'A', the weighted total or mean is computed respectively.

If **stat** = 'N', 'L' or 'S', the only effect of weights is to eliminate values with zero weights from the computations.

*Constraint*: **weight** = 'U', 'V' or 'W'.

4: **isf**(**nfac**) – INTEGER array

Indicates which factors in **ifac** are to be used in the tabulation.

If **isf**$(i) > 0$ the $i$th factor in **ifac** is included in the tabulation.

Note that if **isf**$(i) \leq 0$, for $i = 1, 2, \ldots,$ **nfac** then the statistic for the whole sample is calculated and returned in a $1 \times 1$ table.

5: **lfac**(**nfac**) – INTEGER array

The number of levels of the classifying factors in **ifac**.

*Constraint*: if **isf**$(i) > 0$, **lfac**$(i) \geq 2$, for $i = \mathrm{Ai}, \ldots, \mathrm{Ai}$.

6: **ifac**($ldf$, **nfac**) – INTEGER array

$ldf$, the first dimension of the array, must satisfy the constraint $ldf \geq$ **n**.

The **nfac** coded classification factors for the **n** observations.

*Constraint*: $1 \leq$ **ifac**$(i, j) \leq$ **lfac**$(j)$, for $i = 1, 2, \ldots,$ **n** and $j = 1, 2, \ldots,$ **nfac**.

7: **y**(**n**) – REAL (KIND=nag_wp) array

The variable to be tabulated. If **stat** = 'N', **y** is not referenced.

8: **wt**(:) – REAL (KIND=nag_wp) array

The dimension of the array **wt** must be at least **n** if **weight** = 'W' or 'V', and at least 1 otherwise

If **weight** = 'W' or 'V', **wt** must contain the **n** weights. Otherwise **wt** is not referenced.

*Constraint*: if **weight** = 'W' or 'V', **wt**$(i) \geq 0.0$, for $i = \mathrm{Ai}, \ldots, \mathrm{Ai}$.

9: **table**(**maxt**) – REAL (KIND=nag_wp) array

If **update** = 'U', **table** must be unchanged from the previous call to nag_contab_tabulate_stat (g11ba), otherwise **table** need not be set.

10: **ncells** – INTEGER

If **update** = 'U', **ncells** must be unchanged from the previous call to nag_contab_tabulate_stat (g11ba), otherwise **ncells** need not be set.

11: **icount**(**maxt**) – INTEGER array

If **update** = 'U', **icount** must be unchanged from the previous call to nag_contab_tabulate_stat (g11ba), otherwise **icount** need not be set.

12: **auxt**(:) – REAL (KIND=nag_wp) array

The dimension of the array **auxt** must be at least **ncells** if **stat** = 'A', $2 \times$ **ncells** if **stat** = 'V', and at least 1 otherwise

If **update** = 'U', **auxt** must be unchanged from the previous call to nag_contab_tabulate_stat (g11ba), otherwise **auxt** need not be set.

## 5.2 Optional Input Parameters

1: **n** – INTEGER

*Default*: the dimension of the array **y** and the first dimension of the array **ifac**. (An error is raised if these dimensions are not equal.)

The number of observations.

*Constraint*: $\mathbf{n} \geq 2$.

2: **nfac** – INTEGER

*Default*: the dimension of the arrays **isf**, **lfac** and the second dimension of the array **ifac**. (An error is raised if these dimensions are not equal.)

The number of classifying factors in **ifac**.

*Constraint*: $\mathbf{nfac} \geq 1$.

3: **maxt** – INTEGER

*Default*: the dimension of the arrays **table**, **icount**. (An error is raised if these dimensions are not equal.)

The maximum size of the table to be computed.

*Constraint*: **maxt** $\geq$ product of the levels of the factors included in the tabulation.

## 5.3 Output Parameters

1: **table**(**maxt**) – REAL (KIND=nag_wp) array

The computed table. The **ncells** cells of the table are stored so that for any two factors the index relating to the factor referred to later in **lfac** and **ifac** changes faster. For further details see Section 9.

2: **ncells** – INTEGER

The number of cells in the table.

3: **ndim** – INTEGER

The number of factors defining the table.

4: **idim**(**nfac**) – INTEGER array

The first **ndim** elements contain the number of levels for the factors defining the table.

5: **icount**(**maxt**) – INTEGER array

A table containing the number of observations contributing to each cell of the table, stored identically to **table**. Note if **stat** = 'N' this is the same as is returned in **table**.

6: **auxt**(:) – REAL (KIND=nag_wp) array

The dimension of the array **auxt** will be **ncells** if **stat** = 'A', $2 \times$ **ncells** if **stat** = 'V' and 1 otherwise

If **stat** = 'A' or 'V', the first **ncells** values hold the table containing the sum of the weights for the observations contributing to each cell, stored identically to **table**.

If **stat** = 'V', the second set of **ncells** values hold the table of cell means. Otherwise **auxt** is not referenced.

7: **ifail** – INTEGER

**ifail** $= 0$ unless the function detects an error (see Section 5).

## 6 Error Indicators and Warnings

Errors or warnings detected by the function:

**ifail** $= 1$

On entry, **n** $< 2$,
or **nfac** $< 1$,
or $ldf <$ **n**,
or **update** $\neq$ 'I' or 'U',
or **weight** $\neq$ 'U', 'W' or 'V',
or **stat** $\neq$ 'N', 'T', 'A', 'V', 'L' or 'S'.

**ifail** $= 2$

On entry, **isf**$(i) > 0$ and **lfac**$(i) < 2$, for some $i$,
or **ifac**$(i, j) < 1$, for some $i, j$,
or **ifac**$(i, j) >$ **lfac**$(j)$ for some $i, j$,
or **maxt** is too small,
or **weight** $=$ 'W' or 'V' and **wt**$(i) < 0.0$, for some $i$.

**ifail** $= 3$

**stat** $=$ 'V' and the divisor for the variance is $\leq 0.0$.

**ifail** $= 4$

**update** $=$ 'U' and at least one of **ncells**, **table**, **auxt** or **icount** have been changed since previous call to nag_contab_tabulate_stat (g11ba).

**ifail** $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

**ifail** $= -399$

Your licence key may have expired or may not have been installed correctly.

**ifail** $= -999$

Dynamic memory allocation failed.

## 7 Accuracy

Only applicable when **stat** $=$ 'V'. In this case a one pass algorithm is used as described by West (1979).

## 8 Further Comments

The tables created by nag_contab_tabulate_stat (g11ba) and stored in **table**, **icount** and, depending on **stat**, also in **auxt** are stored in the following way. Let there be $n$ factors defining the table with factor $k$ having $l_k$ levels, then the cell defined by the levels $i_1$, $i_2, \ldots, i_n$ of the factors is stored in the $m$th cell given by

$$m = 1 + \sum_{k=1}^{n} [(i_k - 1)c_k],$$

where $c_j = \prod_{k=j+1}^{n} l_k$, for $j = 1, 2, \ldots, n-1$ and $c_n = 1$.

# 9 Example

The data, given by John and Quenouille (1977), is for a $3 \times 6$ factorial experiment in 3 blocks of 18 units. The data is input in the order, blocks, factor with 3 levels, factor with 6 levels, yield. The $3 \times 6$ table of treatment means for yield over blocks is computed and printed.

## 9.1 Program Text

```
     function g11ba_example

fprintf('g11ba example results\n\n');

ifac = [nag_int(1),1,1; 1,2,1; 1,3,1; 1,1,2; 1,2,2; 1,3,2;
                1,1,3; 1,2,3; 1,3,3; 1,1,4; 1,2,4; 1,3,4;
                1,1,5; 1,2,5; 1,3,5; 1,1,6; 1,2,6; 1,3,6;
                2,1,1; 2,2,1; 2,3,1; 2,1,2; 2,2,2; 2,3,2;
                2,1,3; 2,2,3; 2,3,3; 2,1,4; 2,2,4; 2,3,4;
                2,1,5; 2,2,5; 2,3,5; 2,1,6; 2,2,6; 2,3,6;
                3,1,1; 3,2,1; 3,3,1; 3,1,2; 3,2,2; 3,3,2;
                3,1,3; 3,2,3; 3,3,3; 3,1,4; 3,2,4; 3,3,4;
                3,1,5; 3,2,5; 3,3,5; 3,1,6; 3,2,6; 3,3,6];
y   = [           274;   361;   253;   325;   317;   339;
                  326;   402;   336;   379;   345;   361;
                  352;   334;   318;   339;   393;   358;
                  350;   340;   203;   397;   356;   298;
                  382;   376;   355;   418;   387;   379;
                  432;   339;   293;   322;   417;   342;
                   82;   297;   133;   306;   352;   361;
                  220;   333;   270;   388;   379;   274;
                  336;   307;   266;   389;   333;   353];

weight = 'U';
wt     = [];

stat   = 'A';
update = 'I';
lfac   = [nag_int(3); 3; 6];
isf    = [nag_int(0); 1; 1];
maxt   = prod(lfac(isf~=0));

table  = zeros(maxt,1);
icount = zeros(maxt,1,nag_int_name);
auxt   = zeros(2*maxt,1);
ncells = nag_int(0);

% Compute table
[table, ncells, ndim, idim, icount, auxt, ifail] = ...
  g11ba( ...
         stat, update, weight, isf, lfac, ifac, y, wt, ...
         table, ncells, icount, auxt);

% Display results
fprintf(' Table\n\n');
ncol = idim(ndim);
nrow = ncells/ncol;
table  = transpose(reshape(table,[ncol,nrow]));
icount = transpose(reshape(icount,[ncol,nrow]));
for i = 1:nrow
  row = [table(i,:); double(icount(i,:))];
  fprintf('%8.2f(%2d)', row);
  fprintf('\n');
end
```

## 9.2 Program Results

```
     g11ba example results

 Table

  235.33( 3)  342.67( 3)  309.33( 3)  395.00( 3)  373.33( 3)  350.00( 3)
  332.67( 3)  341.67( 3)  370.33( 3)  370.33( 3)  326.67( 3)  381.00( 3)
  196.33( 3)  332.67( 3)  320.33( 3)  338.00( 3)  292.33( 3)  351.00( 3)
```